

École Polytechnique de Montréal

Département Génie Informatique et Génie Logiciel

INF3710 – Fichiers et Bases de données

TP 3 – Manipulation de bases de données en SQL

Objectif: maîtriser la syntaxe SQL et être capable de formuler des requêtes en SQL, ainsi que de développer des fonctions, des triggers et des vues

1. Informations générales

<i>Pondération</i>	5%
<i>Taille de l'équipe</i>	2 personnes

Notez bien:

1. Tout retard dans la remise du TP entraîne automatiquement une pénalité comme discuté dans le plan de cours.
2. Aucun TP ne sera corrigé, s'il est soumis par une équipe dont la taille est différente **de deux (2) étudiants** sans l'approbation préalable du chargé de laboratoire. Cette approbation ne sera accordée à une équipe qu'en cas de nombre impair d'étudiants dans le laboratoire. Sinon, la note de zéro sera attribuée aux étudiants concernés.
3. Soumission du TP par **Moodle** uniquement (<https://moodle.polymtl.ca>). Aucune soumission "hors **Moodle**" ne sera corrigée. La note de zéro sera attribuée aux étudiants concernés.

1. Évaluation

Rubriques	Points
Créer la base de données et insérer des données	3
Requêtes SQL	10
Vues	4
Triggers	3

Critères d'évaluation:

- Scripts SQL corrects, fonctionnels et complets

PARTIE 1 : TUTORIEL (NON NOTE)

Exercice 1

Soit la base de données **Hotel**. On vous fournit le script SQL correspondant.

Hotel (hotelNo, hotelName, city)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, guestNo, roomNo, dateFrom, dateTo)

Guest (guestNo, guestName, guestAddress)

Formulez les requêtes suivantes :

1. Affichez tous les clients par ordre alphabétique croissant de nom (et décroissant)
2. Lister le nom et l'adresse des clients à Kingston par ordre alphabétique de nom.
3. Affichez les informations des chambres de l'hôtel H111 incluant un l'ancien prix et un nouveau prix incrémenté de 10%. Le nouveau prix soit se trouver dans une colonne nommée « newP »
4. Affichez le nom des hôtels qui sont à London ou Kingston (de deux manières)
5. Affichez le nom des hôtels qui ne sont ni à London ni à Kingston (de deux manières)
6. Lister toutes les réservations pour lesquelles aucune *dateTo* n'a été spécifiée
7. Retrouvez tous les clients féminins dont le nom contient Kat.
8. Affichez les chambres par ordre croissant d'hôtel et par ordre décroissant de prix

Exercice 2

A partir de la base de données **Hotel**, répondez aux questions additionnelles suivantes :

1. Lister les numéros de chambres (et le nom de l'hôtel) qui sont actuellement occupées au Grosvenor Hotel.
2. En utilisant la requête précédente, lister le détail des chambres actuellement inoccupées au Grosvenor Hotel.
3. En utilisant la requête précédente (Lister les chambres qui sont actuellement occupées au Grosvenor Hotel.), quel est le revenu perdu de ces chambres non occupées?
4. Affichez le numéro de chambre, le numéro d'hôtel et le nom de l'hôtel pour les chambres réservées. Utilisez une jointure simple et une jointure naturelle.

Exercice 3

Ecrivez les requêtes SQL pour répondre aux questions suivantes :

1. Combien y a-t-il d'hôtels?
2. Quel est le prix total tiré des chambres doubles?
3. Lister le nombre de chambres dans chaque hôtel et ordonnez-les par ordre croissant en fonction du nombre de chambres
4. Lister le nombre de chambres dans chaque hôtel qui contient plus de 1 chambre ordonné par nombre de chambres en ordre croissant

Exercice 4

Ecrivez les requêtes SQL pour répondre aux questions suivantes :

1. Lister le prix et le type des chambres au Grosvenor Hotel en utilisant une sous-requête
2. Lister le prix et le type des chambres au Grosvenor Hotel en utilisant une jointure
3. Lister les clients qui sont actuellement au Grosvenor Hotel avec des sous-requêtes.
Current_date retourne la date courante en PostgreSQL
4. Lister le nombre de chambres de chaque hôtel à London
5. Effectuez le produit cartésien de Hôtel et Room. Que remarquez-vous?
6. Indiquez les chambres qui sont réservées ainsi que leur hôtel (Proposez deux manières différentes)
7. Lister le prix et le type des chambres au Grosvenor Hotel en utilisant une jointure
8. Affichez les détails de tous les clients qui ont effectué une réservation
9. Affichez les détails de tous les clients qui ont effectué une réservation. Utilisez des alias.
10. Lister tous les hôtels et leurs réservations qu'ils en aient ou pas.

Exercice 5

1. On veut créer une fonction en PL/pgSQL qui retourne le nombre total d'enregistrements dans la table hotel ;
2. Dans votre base de données Hotel, créez un trigger lors de l'insertion d'une nouvelle chambre dans Room pour les situations suivantes:
 - a) Trigger 1 : Le prix des chambres doubles doit être supérieur à 100 \$. Sinon vous devez lancer une exception
 - b) Trigger 2: Le prix des chambres doubles doit être supérieur au prix de la chambre simple la plus coûteuse. Sinon vous devez lancer une exception

PARTIE 2 : TP

Exercice 1 (noté, 20 points)

1. Travail à faire

Soit la base de données suivante (les PK sont soulignées)

Constructeur(idConstructeur, nomCompagnie, type_constructeur , pays)

Avion(idAvion, modele, idConstructeur, nbMoteurs, nbSieges)

FK idConstructeur references Constructeur(idConstructeur)

Moteur(idMoteur, modele, idConstructeur, diametre, longueur, poids, pousse)

FK idConstructeur references Constructeur(idConstructeur)

Flotte(noSerieAvion, noSerieMoteur, idMoteur, idAvion, vitesse, autonomie, dateService, heuresVol, dateRetrait)

FK idMoteur references Moteur(idMoteur)

FK idAvion references Avion (idAvion)

- a) (2 points)** Créer la base de données `Flotte` correspondante. Le script DDL de création de la BD doit se nommer `Flotte_schema.sql`. Vous devez respecter les contraintes suivantes :
- La variable `dateService` est obligatoire, tandis que celle `dateRetrait` peut être vide car on ne connaît pas, au moment de la création de la base de données, la date à laquelle l'avion sera retiré de service.
 - On doit s'assurer que le `type_constructeur` indique un choix parmi une liste pré-établie 'motoriste', 'avionneur' ou 'systemes' seulement.
 - On doit générer automatiquement une valeur automatique dans les champs `idConstructeur`, `idAvion`, `idMoteur` associés respectivement aux tables `Constructeur`, `Avion`, `Moteur`.
- b) (1 point)** Insérer des données dans toutes les tables. Votre script d'insertion des données doit se nommer `Flotte_data.sql` (les données à mettre dans vos tables vous sont communiquées dans le fichier `data.txt`).
- c) (14 points)** Écrire les requêtes SQL suivantes. Votre script de requêtes SQL doit se nommer `Flotte_query.sql`. Toutes vos requêtes doivent être précédées d'un commentaire qui reprend le texte de la requête en Français.
- 1) (1 point)** Retourner le nombre d'appareils produits par pays.
 - 2) (1.5 point)** Retourner le nombre d'appareils produits par pays dont le nombre de sièges est supérieur à 200.
 - 3) (1 point)** Préparer une requête afin de réaliser une analyse de l'état de la flotte à partir des données suivantes : le modèle, le nombre de sièges, le nombre d'heures accumulées et la vitesse.
 - 4) (1.5 point)** Quels sont les avions encore en service ? Proposer une requête en listant l'avion, son modèle, le nombre de sièges et le nombre d'heures de vol.
 - 5) (1 point)** Retourner le nombre de sièges et le nombre d'heures de vol des flottes.
 - 6) (1 point)** Quels sont le ou les modèles d'appareils dont le nombre de sièges est supérieur à 300? Affichez le modèle et le nom de la compagnie.
 - 7) (1.5 point)** Quel est le nombre d'appareil (avions) dont le nombre des sièges est supérieur à la moyenne des sièges de tous les appareils, en utilisant une ou des sous-requêtes?
 - 8) (1.5 point)** Quelle est la différence entre le plus grand et le plus petit diamètre d'un moteur ? Affichez le résultat dans une colonne nommée `différence`.
 - 9) (1.5 point)** Créer une vue `Avion_Bombardier_Vue` ayant le nom de la compagnie, le modèle, le nombre des sièges et le nombre des moteurs d'avion construit par 'Bombardier Aéro'.
 - 10) (0.5 point)** Modifiez la vue en utilisant `CHECK Option`. Que se passe-t-il ?
 - 11) (1 point)** Créez la vue `Avion_Bombardier_Update_Check` qui contient tous les avions de constructeur dont l'id = 1.

- 12) (0.5 points) Mettez à jour les tuples de la vue en 11) en changeant l'id constructeur à 2. Que se passe-t-il ?
- 13) (0.5 points) Modifiez la vue en 11) en rajoutant l'option WITH CHECK OPTION. Ré-exécutez le code de création de la vue et essayez de mettre à jour la vue comme dans 12. Que se passe-t-il ?

d) Triggers (3 points)

Créez un fichier *Flot_trigger_function.sql* qui contient les éléments suivants :

- 1) (0.5 points) Une fonction *nbAvions* en PL/pgSQL qui retourne le nombre total d'avions;
- 2) (2.5 points) Un trigger *contrôle_inspection* qui crée une insertion dans une table nommée *Controle*. Ce trigger se déclenche lorsqu'il y a une mise à jour d'un nouveau modèle d'avion. On veut conserver les informations de l'avion ainsi que la date de la mise à jour.

2. Livrable

Vous devez soumettre sur Moodle à la date d'échéance un fichier *matricule1_matricule2_TP3.zip* qui contient les fichiers suivants :

- a) Le fichier *matricule1_matricule2_Flotte_schema.sql*,
- b) Le fichier *matricule1_matricule2_vol_Flotte_data.sql*,
- c) Le fichier *matricule1_matricule2_vol_Flotte_query.sql*,
- d) Le fichier *matricule1_matricule2_Flot_trigger_function.sql*.